KHEPERA III STARGAZER

USER MANUAL

VERSION 1.1
JANUARY 2013

Documentation Author

Julien Tharin
K-Team S.A.
Z.I. Plans-Praz
1337 Vallorbe
Switzerland

Email: info@k-team.com
Url: www.k-team.com

Documentation version

Version	Date	Author	Description
1.0	08.05.2012	J. Tharin	First draft
1.1	31.01.2013	J. Tharin	LRF/battery card ON update

Trademark Acknowledgements:

IBM PC : International Business Machines Corp.
Macintosh : Apple Corp.
SUN Sparc-Station : SUN Microsystems Corp.
LabVIEW : National Instruments Corp.
Matlab : MathWorks Corp.
Webots : Cyberbotics Ltd
Logitech : Logitech Int. SA
Gumstix : Gumstix Inc.
Khepera : K-Team SA
Stargazer : Hagisonic co. Ltd

LEGAL NOTICE:

- The contents of this manual are subject to change without notice
- All efforts have been made to ensure the accuracy of the content of this manual. However, should any error be detected, please inform K-Team.
- The above notwithstanding, K-Team can assume no responsibility for any error in this manual.

TABLE OF CONTENTS

1. INTRODUCTION 1

1.1 How to use this handbook 1
1.2 Safety precautions 2
1.3 Recycling 2
1.4 Specifications 3

2. UNPACKING AND INSPECTION 4

2.1 Package Contents 4
2.2 Inspection 4

3. DESCRIPTION 5

3.1 Overview 5
3.2 KSG Hardware 6
3.3 KSG Software 6

4. USAGE 7

4.1 Required hardware / software 7
4.1.1 Required hardware 7
4.1.2 Required software 7
4.2 Assembly 8
4.2.1 KSG on the robot 8
4.2.2 Landmarks setting 9
4.3 Power-up and test 10
4.3.1 Test 10
4.4 Programming the KSG 12
4.4.1 Installation of the latest libkorebot and configuration file 12
4.4.2 Using the KSG with kgazer_test software 14
4.4.3 Compiling your own program using the libkorebot 15
4.4.4 Software example 22

5. WARRANTY 24

1. INTRODUCTION

Thank you for buying the Khepera III Stargazer (abbreviated KSG in this document)!

With this module, you will be able to have global position and bearing of your Khepera3 robot.

1.1 How to use this handbook

This handbook introduces the KSG and its various operating modes. For a quick start, jump to chapter 4 "***Usage***".

If this handbook does not answer one of the problems you wish to solve, please consult the K-Team web site (http://www.k-team.com) and especially the Forum and the FAQs.

- **Unpacking and Inspection** : KSG package description and first use
- **Description** : KSG description
- **Usage** : KSG usage descriptions.
- **Warranty** : Legal notice of the KSG warranty.

1.2 Safety precautions

Here are some recommendations on how to correctly use the KSG:

- **Keep the board away from wet area**. Contact with water could cause malfunction and/or breakdown.
- **Store your board in a stable position**. This will avoid the risks of falling, which could break it or cause damage to a person.
- **Do not plug any connectors while the board is powered on**. To avoid any damage, make all connections when the board power is off.
- **Never leave the KSG powered when it is unused**. When you have finished working with KSG, turn it off. It will save the battery life.

1.3 Recycling

Think about the end of life of your product! Parts of the board can be recycled and it is important to do so. By recycling you can help to create a cleaner and safer environment for generations to come. For those reasons please take care to the recycling of your product at the end of its life cycle, for instance sending back the product to the manufacturer or to your local dealer.

Thanks for your contribution to a cleaner environment!

1.4 Specifications

The main specifications of the KSG card are listed below:

- Range max : 2.5-3 m of diameter with 1 landmarks (for ceiling height of 2.4 m*[1])
- Accuracy : 3.5 cm / 3 degrees *[2]
- Measurement time : 10 times /sec
- Autonomy : 2h30 ; Li-Pol battery
- Size : 85 width x 80 depth*[3] x 56 height*[4] [mm]
- Mass : 117 g
- Compatibility : Khepera III extensions (KB-250 specifications)

*1 ceiling height can be from 1.1 to 6.0 m but with different landmarks; range may vary

*2 with a homogenously flat floor; may increase with an extension (Khepera3-Gripper)

*3 with cables

*4 without lower ERNI connectors (as plugged)

2. UNPACKING AND INSPECTION

2.1 Package Contents

Figure 2-1: Contents of the KSG Pack

Your package should contain the following items:

1. KSG board
2. DVD with software and this User Manual*

* Updates can be found at http://www.k-team.com/

2.2 Inspection

First check that you have a complete package. You should find:

- the KSG module
- CD with
  - This user manual
  - The Libkorebot library software

Then check that the wires connecting the Stargazer to the battery module are connected and not in bad shape.

3. Description

3.1 Overview

An overview of the KSG hardware is depicted in the Figure 3.1. The locations of various key elements are indicated for later references.

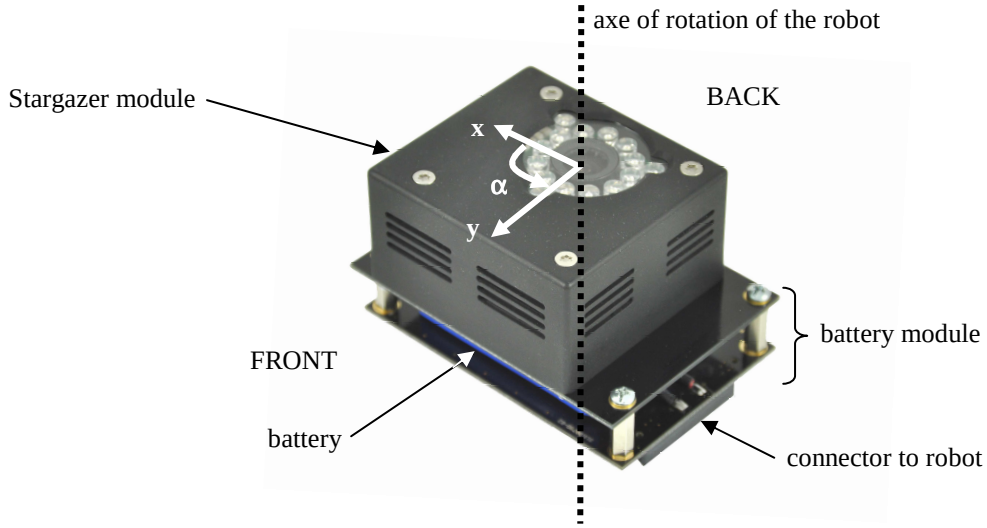


Figure 3.1 KSG overview

3.2 KSG Hardware

The KSG module is composed of a Stargazer position module from Hagisonic co. LTD and the K3LFSG battery module. There are also landmarks to put on the ceiling.

The Stargazer module has IR Led's for lighting landmarks and a camera for capturing the image, which is processed by onboard electronics. It computes position and bearing related to landmarks stuck on the ceiling.

The K3LFSG battery module is composed of a battery and electronics for managing its charge and connections to the robot. The module is ON at startup, and OFF when the Khepera III/Korebot II is OFF.

3.3 KSG Software

The position and bearing are computed internally in the Stargazer and they are communicated through the serial port (/dev/ttyS2) to the robot using several commands.

For the ease in use, functions were developed and included in the Libkorebot library. A test program named ***kgazer_test*** is also provided. It initializes the module and displays the position in an ASCII x-y graph. The robot can be controller with the keyboard arrows.

The parameters of the KSG are saved into the KSG and remain even after power OFF/ON. The exception is the calibration parameter.

4. Usage

4.1 Required hardware / software

The required hardware and software to use the board and develop programs are described below.

4.1.1 Required hardware

- Computer with Bluetooth or Wifi access
- KSG module
- Khepera 3 with Korebot II plugged

4.1.2 Required software

- Korebot 2 light toolchain installed (see Korebot 2 User Manual)
- Libkorebot library (from version 1.19; see chapter 4.4.1)

Remarks: you may find updated version of these software at:
http://ftp.k-team.com/KorebotII/software/common/libkorebot/

4.2 Assembly

4.2.1 KSG on the robot

The assembly of the board with the robot is depicted in Figure **4-1**. The Korebot 2 must be already in the robot and its upper part plugged on top (see “Khepera 3 User Manual”, chapter 3.1.7).

- Insert the KSG module on top of the Korebot 2 upper board.
- Pay attention to the orientation of the module (front/back).
- The assembly and disassembly must be done while the robot is switched OFF; also the robot and KSG must not be charging.

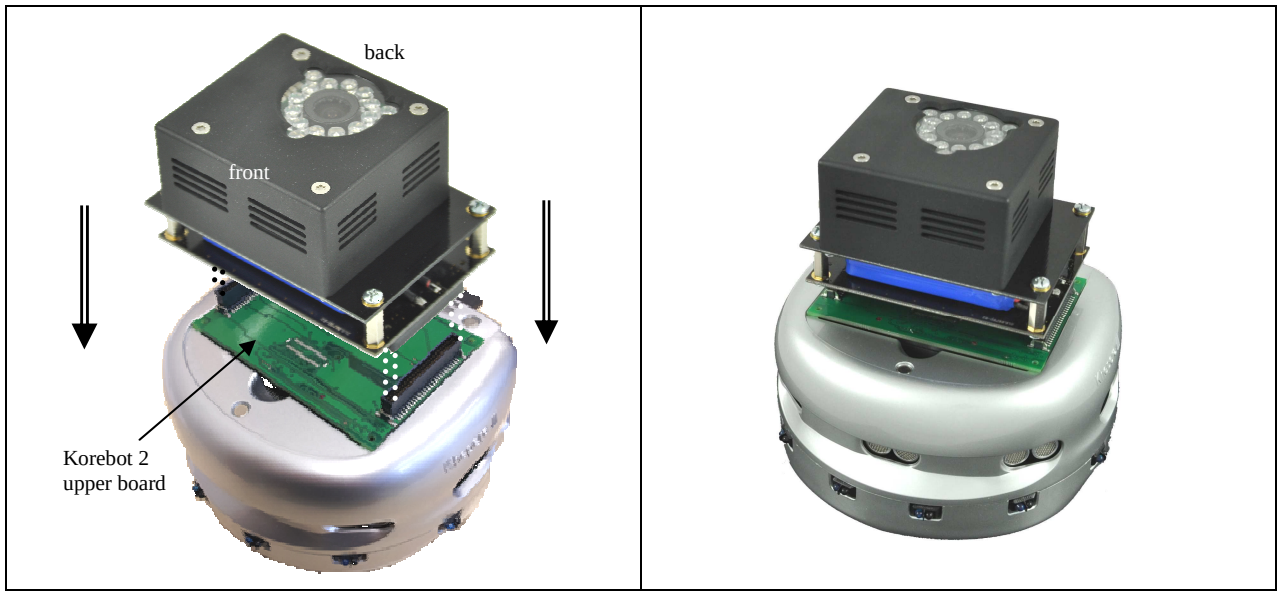


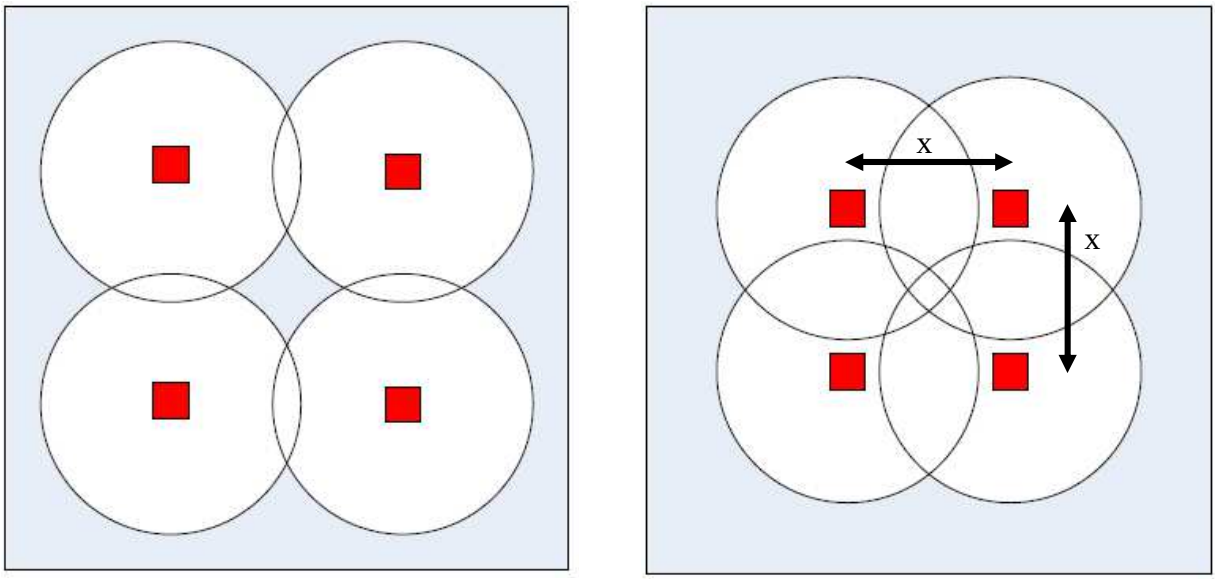
Figure 4-1: Basic assembly

4.2.2 Landmarks setting

The landmarks should be placed at maximum of ***x*** m interval on the ceiling for the height of about ***h*** m in order that any dead zone may not occur (figure 4-2 b), with a relation of:

Equation 4.1: $x = 0.8 \cdot h$

Example: h=2.5 m => x = max 2m



a) with dead zone (b) without dead zone

Figure 4-2: The placement of landmarks

You don’t need to install all the 4 landmarks if your area is smaller. And they don’t need to be placed in square shape because during the map building process, the relative position of the landmarks are computed.

The default landmarks delivered with the KSG are of type ***HLD1S***. They can be used with a height range of 1.1-2.9 m. See chapter 3 A and appendix C of “Stargazer User’s Guide” for more information about the landmark:

http://ftp.k-team.com/KheperaIII/KSG/StarGazer_Guide_02.0904.16(English).pdf

4.3 Power-up and test

The KSG module is switched ON when the Khepera 3 robot is switched ON. It starts autonomously sending position through the serial port, if it sees a landmark.

4.3.1 Test

You can also launch the ***kgazer_test*** program (see chapter 4.4.2) or test with the instructions below:

- put the robot so that it can see landmarks (for placing landmarks, see chapter 4.2.2).
- after the robot is switched ON, you should see some red LED ON inside the Stargazer, from its top window.
- change ssh escape character on your PC . If it is ~, it will conflict only with the Stargazer command for the test below, not for the normal use of that manual for Linux, run:

  sudo sh -c 'echo "EscapeChar ^" >>/etc/ssh/ssh_config'
- open a ssh connections terminal to the robot (see ***Korebot II User Manual***, chapter 5.2.6).
- run the following command to set the serial port the terminal:

  stty -F /dev/ttyS2 115200 -parenb cs8 clocal -crtscts

  cat /dev/ttyS2

  => it should display lines like without interruption if the robot is under landmarks on the first one:

  ^I560|+79.05|-10.94|-106.12|190.00`~^I560|+79.05|-10.93|-106.11|190.00`~^I560|+78.39|-10.97|-106.15|189.16`~^I560|+79.05|-10.94|-106.15|190.00`~^I560|+79.05|-10.94|-106.13|190.00`~^I560|+79.05|-10.93|-106.12|190.00`~^I560|+79.05|-10.93|-106.11|189.82`

  The data are respectively landmark id (560), angle (~79 deg), x position (~-10cm), y position (~-106cm), height to landmark (~190cm).

  To stop, push keys ***CTRL*** and ***c*** together at the first terminal.

To test sending commands, you must install minicom on your Korebot II:

- copy these following files to the Korebot II and install then with the synthax: ***ipkg install FILENAME.ipk***

http://ftp.k-team.com/KorebotII/software/common/ncurses_5.4-r9_armv5te.ipk

http://ftp.k-team.com/KorebotII/software/common/minicom_2.1-r0_armv5te.ipk

- launch minicom with the command: ***minicom -o***
- set its parameters with the sub-menu "**Serial port**" setup of the menu **[configuration]** (Figure **4.4**) (keys "**Ctrl-a + o**") as described in Figure 4.3.

```
+-----------------------------------------------------------------+
| A - Serial Device : /dev/ttyS2                                  |
| B - Lockfile Location : /var/lock                               |
| C - Callin Program :                                            |
| D - Callout Program :                                           |
| E - Bps/Par/Bits : 115200 8N1                                   |
| F - Hardware Flow Control : No                                  |
| G - Software Flow Control : No                                  |
|                                                                 |
| Change which setting?                                           |
+-----------------------------------------------------------------+
```

Figure 4.3: Minicom serial parameters

- Save the settings with the sub-menu "***Save setup as dfl***" (Figure 4.4).

```
+-----[configuration]------+
| Filenames and paths      |
| File transfer protocols  |
| Serial port setup        |
| Modem and dialing        |
| Screen and keyboard      |
| Save setup as dfl        |
| Save setup as..          |
| Exit                     |
+--------------------------+
```

Figure 4.4: Minicom configuration menu

- push ***CTRL a*** and ***e*** keys to have local echo.
- write the following command to stop receiving position: ***~#CalcStop`***

  => return should be: ***~!CalcStop`***

  Multiple stops may be needed. The command ***~#CalcStart`*** starts again the position computation.

- write the following command to get Stargazer firmware version: ***~#Version`***

  => return should be: ***~!Version`~$Version|2.1101.18`***

See ***Stargazer User's Guide*** for more information on chapter 6 for commands and on chapter 7 for landmarks:

http://ftp.k-team.com/KheperaIII/KSG/StarGazer_Guide_02.0904.16(English).pdf

4.4 Programming the KSG

The KSG is an extension that can be used only with the KoreBotII mounted in the KheperaIII. That means that the KoreBotII controls all the functions of the KSG. As all the KoreBotII extensions, a library including all the available functions is provided with the Libkorebot version 1.19 or greater. If you already have a Libkorebot installed in your computer but with an older version, you can download the latest version with the following link:

http://ftp.k-team.com/KorebotII/software/common/libkorebot/libkorebot-*VERSION*.zip

(Replace the VERSION by the version number, 1.19 or greater)

If your KoreBotII has already the latest Libkorebot) installed, jump to section 4.4.2.

You can check if the last version is installed:

- Log on the KoreBotII (via ssh, Bluetooth or serial port)
- You can check if you have the 1.19 version in listing the present files:

  ls –s /usr/lib/libkorebot*

  ⇨ This should give this, where the number after ***.so.*** is the current version:
  0 /usr/lib/libkorebot.so 315 /usr/lib/libkorebot.so.1.19

4.4.1 Installation of the latest libkorebot and configuration file

To use the KSG with the KheperaIII and KoreBotII, it's necessary to install the libkorebot version 1.19 or greater on the KoreBotII.

Normally, if you have received the KoreBotII at the same time than the KSG, the KoreBotII is ready to be used with the KSG; in this case, you can jump to section 4.4.2. Otherwise, if you have bought the KSG separately, you will need to execute the step described below:

- Log on the KoreBotII (via ssh, Bluetooth or serial port)

You have different ways to update the library:

Autonomously with the package:

- Upload the package file ***libkorebot-1.19-r0_armv5te.ipk*** by ssh, Bluetooth or Serial to the Korebot II.
- Remove the old one with the command: ***ipkg remove libkorebot***
- Install the new one with the command:

  ipkg install libkorebot-1.19-r0_armv5te.ipk

By hand:

- Remove the old libkorebot: ***rm /usr/lib/libkorebot****
- Copy or send the ***libkorebot.so.1.19*** file (locate in the ***build-korebot-2.6/lib/*** directory of the libkorebot source files) to the directory ***/usr/lib***.
- Link the libkorebot file as ***libkorebot.so****:*

  ln -s /usr/lib/libkorebot.so.1.19 /usr/lib/libkorebot.so

4.4.2 Using the KSG with kgazer_test software

The ***kgazer_test*** is a test and example program given also in source code in directory ***src/tests/*** of the Libkorebot library.

After copying this program to the Korebot II and launching it with the command ***./kgazer_test*** , it will initialize the Stargazer, display the version of its internal software, ask if you would like to change its parameters (number, type and reference of landmark) . Then it will rotate to do a calibration due to the fact that the ceiling may not exactly parallel to the KSG module. Then it displays an ASCII interface (figure 4.5).

You can move the robot with the keyboard arrows keys. The robot is represented by the character ***A***,<,> or ***V*** depending of its orientation. Other keys are defined below:

+/-	: zoom in / out
Page Up / Down	: speed up/down the robot
c	: redo position calibration
g	: goto xy (choose goal, then go to goal using a very basic algorithm)
m	: if several landmarks, start map building
k	: use / don't use correction from calibration
p	: change stargazer parameters
s	: start/stop saving data to file *data_corr.csv*
t	: display / undisplay old robot position
q	: exit program

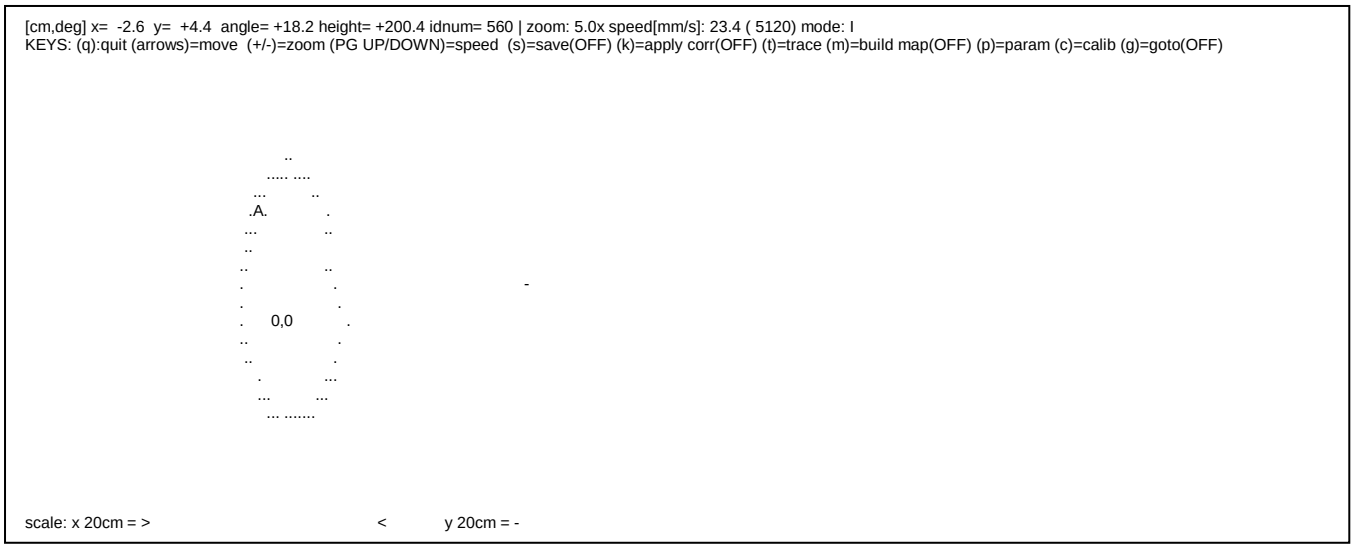


Figure 4.5: kgazer_test ASCII interface

4.4.3 Compiling your own program using the libkorebot

To make your own software to control the KSG, you need to install the development tools for the KoreBotII on your computer (with Linux OS). If it is not already done, please look at the KoreBotII User Manual to know how to install it.

Once done, you can start writing your program. The best solution is to copy the ***kgazer_small_ex.c*** source code (see chapter 4.4.4) and modify it. The source code ***kgazer_test.c*** is a more complete example. In any case, keep your program in the ***src/tests/*** directory, and modify the Makefile to compile your new software: add your new program filename in line 35 at the variable ***TARGETS*** of the file in the *src/test/Makefile* or start with the template program in the *template* directory.

As for all the KoreBotII extensions, you need to initialize the library before using the functions described below.

The Libkorebot provides high-level functions which include all the available capabilities of the KSG. Look at the following sections to view the different exported constants variables and functions provided by the Libkorebot for that KSG extension.

4.4.3.1 Constants

Constants defined in the header file are described below:

NB_MARK_TYPES

Number of types of different landmarks:

#define NB_MARK_TYPES 6

HLD1S

Index of landmark type HLD1S:

#define HLD1S 0

HLD1L

Index of landmark type HLD1L:

#define HLD1L 1

HLD2S

Index of landmark type HLD2S:

#define HLD2S 2

HLD2L

Index of landmark type HLD2L:

#define HLD2L 3

HLD3S

Index of landmark type HLD3S:

#define HLD3S 4

HLD3L

Index of landmark type HLD3L:

#define HLD3L 5

NB_MARK_MODES

Number of landmark modes:

#define NB_MARK_MODES 2

MARK_ALONE

Alone landmark mode index:

#define MARK_ALONE 0

MARK_MAP

Map landmark mode index:

#define MARK_MAP 1

NB_HEIGHT_FIX_MODES

Number of landmark height modes:

#define NB_HEIGHT_FIX_MODES 2

HEIGHT_FIX_NO

Index of the non-fix height:

#define HEIGHT_FIX_NO 0

HEIGHT_FIX_YES

Index of the fixed height:

#define HEIGHT_FIX_YES 1

ANGLE_CORRECTION

Angle correction for calibration:

#define ANGLE_CORRECTION 137.0

CALIB_STDEV_MAX

Maximum standard deviation before calibration error [cm]:

#define CALIB_STDEV_MAX 3.0

4.4.3.2 Variables

External variables for transferring data that are declared in the header file and defined in the library are described below:

kb_gazer_landmark_types

Array of array of char containing landmark types, defined like this:

*const char ***kb_gazer_landmark_types***[] =*
{"HLD1S","HLD1L","HLD2S","HLD2L","HLD3S","HLD3L"};

kb_gazer_landmark_modes

Array of array of char containing landmark modes, defined like this:

*const char ***kb_gazer_landmark_modes***[] = {"Alone","Map"};*

kb_gazer_height_fix_modes

Array of array of char containing height modes, defined like this:

*const char ***kb_gazer_height_fix_modes***[] = {"No","Yes"};*

4.4.3.3 High-level functions of the libkorebot

There below you will find details about declared functions for the KSG extensions in the Libkorebot libraries.

The **green** words indicate the type of the function (or parameter) and must not be included in your code.
The **blue** words are the parameters. You must declare it in your code before calling the function.
Only the ***black*** words can be directly added to your code.

int *kb_stargazer_Init (char * DeviceName)*

Initialize the library to use the different KSG functions and stop the module sending position. This function must be called at the beginning of each program using the KSG.

Return: *0* : no error
-1 : error initialising gpio
-2 : cannot open serial port
-3 : cannot communicate with the Stargazer

Parameter: **** DeviceName***: name of the device where the KSG is connected. Should be "/dev/ttyS2".

int *kb_gazer_get_version (char * version)*

Get Stargazer firmware version.

Return: *0* : no error
-1 : serial port not open
-2 : command not acknowledged
-3 : command not acknowledged
Parameter: **** version***: firmware version.

int *kb_gazer_set_landmark_number (int number)*

Set the numbers of landmarks to be used.

Return: *0* : no error
-1 : serial port not open
-2 : command not acknowledged
Parameter: ***number***: numbers of landmarks.

int *kb_gazer_get_landmark_number (int * number)*

Get the numbers of landmarks to be used.

Return: *0* : no error
-1 : serial port not open
-2 : command not acknowledged
Parameter: **** number***: numbers of landmarks.

int *kb_gazer_set_ref_if (int refid)*

Set the landmark id as reference.

Return: *0* : no error
-1 : serial port not open
-2 : command not acknowledged
Parameter: ***refid***: reference id.

int *kb_gazer_get_ref_if (int * refid)*

Get the landmark id as reference.

Return: *0* : no error
-1 : serial port not open
-2 : command not acknowledged
Parameter: ****refid***: reference id.

int *kb_gazer_set_landmark_type (int type)*

Set the landmark type index.

Return: *0* : no error
-1 : serial port not open
-2 : command not acknowledged
-3 : type not valid
Parameter: ***type***: landmark type index.

int *kb_gazer_get_landmark_type (int * type)*

Get the landmark type index.

Return: *0* : no error
-1 : serial port not open
-2 : command not acknowledged
-3 : unknown landmark type
Parameter: ***type***: landmark type index.

int *kb_gazer_set_landmark_mode (int mode)*

Set the landmark mode index.

Return: *0* : no error
-1 : serial port not open
-2 : command not acknowledged
-3 : mode not valid
Parameter: ***mode***: landmark mode index.

int *kb_gazer_get_landmark_mode (int * mode)*

Set the landmark mode index.

Return: *0* : no error
-1 : serial port not open
-2 : command not acknowledged
-3 : unknown landmark mode
Parameter: ***mode***: landmark mode index.

int *kb_gazer_set_height_fix_mode (int mode)*

Set the height fix mode index.

Return: *0* : no error
-1 : serial port not open
-2 : command not acknowledged
-3 : mode not valid
Parameter: ***mode***: height fix mode index.

int *kb_gazer_get_height_fix_mode (int * mode)*

Get the height fix mode index.

Return: *0* : no error
-1 : serial port not open
-2 : command not acknowledged
-3 : unknown height fix mode
Parameter: ***mode***: height fix mode index.

int *kb_gazer_start_map_mode (void)*

Start the map building mode

Return: *0* : no error
-1 : serial port not open
-2 : could not stop receiving position
-3 : command not acknowledged
Parameter: -

int *kb_gazer_set_end_command (void)*

Set end of commands for update.
Return: *0* : no error
-1 : serial port not open
-2 : command not acknowledged
-3 : data not upated
Parameter: -

int *kb_gazer_start_computation (void)*

Start computation of position.
Return: *0* : no error
-1 : serial port not open
-2 : command not acknowledged
Parameter: -

int *kb_gazer_wait_stop_computation(void)*

Stop computation of position and wait until it stops (retry MAX_STOP).
Return: *0* : no error
-1 : serial port not open
-2 : command not acknowledged
Parameter: -

int *kb_gazer_stop_computation(void)*

Send the computation stop of position.
Return: *0* : no error
-1 : serial port not open
-2 : command not acknowledged
Parameter: -

void *kb_stargazer_Close (void)*

Release the Stargazer.
Return: -
Parameter: -

int *kb_stargazer_read_data* (*double* * x, *double* * y, *double* * z, *double* * angle, *int* * idnum, *char* * cmode, int corr)

Read and interpret data from the Stargazer. Should be called periodically (up to 10 times/s). ***kb_gazer_start_computation*** must be called once before.

Return:
- *0* : no error
- *-1* : cannot serial port not open
- *-2* : cannot communicate with the Stargazer
- *-3* : buffer overrun (try to call more often this function!)
- *-4* : read command not acknowledged
- *-5* : data error
- *-6* : no landmark found
- *-7* : data error
- *-8* : no received data
- *-9* : mapid error
- *1* : update parameters after map building mode
- *>1*: MAPID in map building mode

Parameter:
- ***x**: x position relative to the reference landmark in [cm], right direction
- ***y**: y position relative to the reference landmark in [cm], forward direction
- ***z**: height to the landmark in [cm]
- ***angle**: angle relative to the reference landmark orientation in [degree] (counterclockwise) in 0..360 range
- ***idnum**: id number of the currently used landmark
- ***cmode**: current mode: 'F' = map building mode, 'I' map mode, 'Z' height calculation mode
- **corr:** 1 apply position correction; 0 do not apply it

int *kb_gazer_calibration* (*knet_dev_t* * mot1,*knet_dev_t* * mot2,*double* *_center_x0,*double* *_center_y0,*double* *_angle_rot,*double* *_a_axis, *double* *_b_axis, *double* *_stddev_x, *double* *_stddev_y)

Configure the Stargazer rotation compensation by moving the robot around itself then fitting the result ellipse.

Return:
- *0* : no error
- *-1*: timeout while computing calibration
- *-2*: data buffer too short
- *-3*: error computing ellipse parameters
- *-4*: error: data are too scattered
- *< -4+ (return kb_stargazer_read_data)*: error getting Stargazer data

Parameter:
- ***mode**: height fix mode index.
- ***mot1**: left motor pointer
- ***mot2**: right motor pointer
- ***_angle_rot**: angle of rotation of the fitted ellipse
- ***_a_axis**: half major axis of the fitted ellipse
- ***_b_axis**: half minor axis of the fitted ellipse
- ***_stddev_x**: standard deviation error of x
- ***_stddev_y**: standard deviation error of y

4.4.4 Software example

A small example of software controlling the KSG is enclosed below. You can find its source code file in ***libkorebot-VERSION/src/tests/kgazer_small_ex.c***.

This sample example does not use calibration of the sensor due to parallelism error of the ceiling and sensor plan. This results in x/y error when the robot rotates. See ***libkorebot-VERSION/src/tests/kgazer_test.c*** for a full example with calibration.

```
#include <korebot/korebot.h>

int main(int argc, char *argv[])
{
   char version[128];
   int ret=0,c=0,i,idnum;
   float angle,x,y,z,xc,yc; // stargazer returned variables and corrected position
   char cmode;     // current mode of the Stargazer

   /* reset the screen */
   kb_clrscr();

   printf("\nKhepera III Stargazer small example program\n");

   /* Set the libkorebot debug level - Highly recommended for development. */
   kb_set_debug_level(2);

   // initialise the libkorebot
   if((ret = kb_init( argc , argv )) < 0 )
     return -1;

   printf("\nInitialising Stargazer module; please wait!\n");

   // initialise the Stargazer module
   if ((ret=kb_stargazer_Init())!=0)
   {
     printf("\nError initialising the Stargazer  (error = %d)!\n",ret);
     kb_stargazer_Close();
     return -2;
   }

   // get Stargazer firmware version
   kb_gazer_get_version(version);
   printf("\nStargazer version is: %s\n",version);

   // read parameters values
   printf("\nRead current Stargazer parameters:\n");
   kb_gazer_get_landmark_number(&c);
   printf("  landmark number: %d\n",c);

   kb_gazer_get_ref_id(&c);
   printf("  reference id   : %d\n",c);

   kb_gazer_get_landmark_type(&c);
   printf("  landmark type  : %s\n",kb_gazer_landmark_types[c]);

   kb_gazer_get_landmark_mode(&c);
   printf("  landmark mode  : %s\n",kb_gazer_landmark_modes[c]);

   kb_gazer_get_height_fix_mode(&c);
   printf("  height fix mode: %s\n",kb_gazer_height_fix_modes[c]);

   kb_gazer_start_computation(); // start computation of position


  // continue on next page
```

```
  for (i=0;i<10; i++) // read 10 data
  {
     // read sensor values
      ret=kb_stargazer_read_data(&x,&y,&z,&angle,&idnum,&cmode,0);

      switch(ret)
      {
         case 0:
               printf("data %d: [cm,deg] x= %+6.1f  y= %+6.1f  angle= %+6.1f height= %+6.1f idnum= %4d mode:
%c\n",i,x,y,angle,z,idnum,cmode);
         break;
         case -3:
            fprintf(stderr,"\nERROR: read error: buffer too short, leaving!\n");
         break;
         case -6:
            printf("The sensor does not see any landmark!\n");
         break;
         case -8:
            fprintf(stderr,"\nERROR: no data received!\n");
         break;
         default:
            fprintf(stderr,"\nERROR: read error number %d!\n",ret);
      }

      usleep(100000); // wait for next data
   }

   kb_stargazer_Close();
   return 0;
}
```

5. WARRANTY

K-TEAM warrants that the KSG is free from defects in materials and workmanship and in conformity with the respective specifications of the product for the minimal legal duration, respectively one year from the date of delivery.

Upon discovery of a defect in materials, workmanship or failure to meet the specifications in the Product during the afore mentioned period, Customer must request help on K-Team Internet forum on http://www.k-team.com/forum/ by detailing:

- the type of KSG used (version)
- the kernel version of the KSG
- the programming environment of the KSG/robot (standard, version, OS)
- the standard use of Product before the appearance of the problem
- the description of the problem.

If no answers have been received within two working days, Customer can contact K-TEAM support by phone or by electronic mail with the full reference of its order and KSG serial number.

K-TEAM shall then, at K-TEAM's sole discretion, either repair such Product or replace it with the equivalent product without charging any technical labor fee and repair parts cost to Customer, on the condition that Customer brings such Product to K-TEAM within the period mentioned before. In case of repair or replacement, K-TEAM may own all the parts removed from the defective Product. K-TEAM may use new and/or reconditioned parts made by various manufacturers in performing warranty repairs and replacement of the Product. Even if K-TEAM repairs or replaces the Product, its original warranty term is not extended.

This limited warranty is invalid if the factory-applied serial number has been altered or removed from the Product.

This limited warranty covers only the hardware and software components contained in the Product. It does not cover technical assistance for hardware or software usage and it does not cover any software products contained in the Product. K-TEAM excludes all warranties expressed or implied in respect of any additional software provided with Product and any such software is provided "AS IS" unless expressly provided for in any enclosed software limited warranty. Please refer to the End User License Agreements included with the Product for your rights with regard to the licensor or supplier of the software parts of the Product and the parties' respective obligations with respect to the software.

This limited warranty is non-transferable.

It is likely that the contents of Customer's flash memory will be lost or reformatted in the course of the service and K-TEAM will not be responsible for any damage to or loss of any programs, data or other information stored on any media or any part of the Product serviced hereunder or damage or loss arising from the Product not being available for use before, during or after the period of service provided or any indirect or consequential damages resulting therefore.

IF DURING THE REPAIR OF THE PRODUCT THE CONTENTS OF THE FLASH MEMORY ARE ALTERED, DELETED, OR IN ANY WAY MODIFIED, K-TEAM IS NOT RESPONSIBLE WHATEVER. CUSTOMER'S PRODUCT WILL BE RETURNED TO CUSTOMER CONFIGURED AS ORIGINALLY PURCHASED (SUBJECT TO AVAILABILITY OF SOFTWARE).

Be sure to remove all third parties' hardware, software, features, parts, options, alterations, and attachments not warranted by K-TEAM prior to Product service. K-TEAM is not responsible for any loss or damage to these items.

This warranty is limited as set out herein and does not cover, any consumable items (such as batteries) supplied with the Product; any accessory products which is not contained in the Product; cosmetic damages; damage or loss to any software programs, data, or removable storage media; or damage due to (1) acts of God, accident, misuse, abuse, negligence, commercial use or modifications of the Product; (2) improper operation or maintenance of the Product; (3) connection to improper voltage supply; or (4) attempted repair by any party other than a K-TEAM authorized module service facility.

This limited warranty does not apply when the malfunction results from the use of the Product in conjunction with any accessories, products or ancillary or peripheral equipment, or where it is determined by K-Team that there is no fault with the Product itself.

K-TEAM EXPRESSLY DISCLAIMS ALL OTHER WARRANTIES THAN STATED HEREINBEFORE, EXPRESSED OR IMPLIED, INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE TO THE FULLEST EXTENT PERMITTED BY LAW.

Limitation of Liability: IN NO EVENT SHALL EITHER PARTY BE LIABLE TO THE OTHER FOR ANY INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES RESULTING FROM PERFORMANCE OR FAILURE TO PERFORM UNDER THE CONTRACT, OR FROM THE FURNISHING, PERFORMANCE OR USE OF ANY GOODS OR SERVICE SOLD OR PROVIDED PURSUANT HERETO, WHETHER DUE TO A BREACH OF CONTRACT, BREACH OF WARRANTY, NEGLIGENCE, OR OTHERWISE. SAVE THAT NOTHING HEREIN SHALL LIMIT EITHER PARTY'S LIABILITY FOR DEATH OR PERSONAL INJURY ARISING FROM ITS NEGLIGENCE, NEITHER PARTY SHALL HAVE ANY LIABILITY TO THE OTHER FOR INDIRECT OR PUNITIVE DAMAGES OR FOR ANY CLAIM BY ANY THIRD PARTY EXCEPT AS EXPRESSLY PROVIDED HEREIN.

K-TEAM S.A.
Z.I PLANS-PRAZ
1337 VALLORBE
SWITZERLAND